REMARKS/ARGUMENTS

The claims have been amended as set forth above. Applicants believe that the amendments to the claims include allowable subject matter. Applicants respectfully request reconsideration of the claims. Applicants reserve the right to pursue the subject matter in any of the cancelled claims in any forthcoming continuation application(s).

I.   Examiner Interview Dated April 30, 2009

An interview was held on April 30, 2009. Applicants believe that an agreement has been reached that the claims are now allowable. A notice of allowance is respectfully requested.

II.   Rejection Under 35 U.S.C. § 101

Claims 10-12, 14, 16 and 17 are rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. The claims have been amended as set forth above to recite "computer-readable storage medium." The storage medium is described in the specification starting at page 4, line 27. Applicants assert that the rejection has been overcome.

III.   Rejection Under 35 U.S.C. § 103

Claims 1, 6-8, 10, 12, 14, and 16-21 are rejected under 35 U.S.C. § 103(a) as being unpatentable over IJDAR, published: November 7, 2000, pages 6-12, by Altamura et al. (hereinafter "Altamura") in view "Star Office XML File Format Working Draft", pages 19, 48, 49, 51, 54-58, 89, 142, and 234, published: January 2001 by Sun Macro (hereinafter "Sun Macro"), and further in view of XML.com, published June 8, 2001, pages 1a and 1, by Eisenberg (hereinafter "Eisenberg") and further in view of "Weak Validation" authored by Jelliffe (hereinafter "Jelliffe"). Claims 9, 11 and 22 are rejected under 35 U.S.C. § 103(a) as being unpatentable over IJDAR, published: November 7, 2000, pages 6-12, by Altamura; "Star Office XML File Format Working Draft", pages 19, 89, 142, and 234, published: January 2001 by Sun Macro; XML.com, published June 8, 2001, pages 1a and 1, by Eisenberg; DFKI, published September 25, 2000, pages 1a, 3, 4, and 11, in view of Jelliffe and further in view of U.S. Patent No. 6,725,426 issued to Pavlov (hereinafter "Pavlov"). In light of the above amendments and

the remarks herein, applicants respectfully disagree with the rejections. Independent claim 1 includes the following combination of features that is not taught or otherwise suggested by the cited references:

> determining properties corresponding to a mini-document of at least one section of *a word processing application document generated on a word processing application*, wherein the mini-document includes a body portion, wherein the mini-document includes at least one member of a group comprising: a header and a footer;
>
> *individually mapping the properties of the mini-document* into a markup language element that is stored *with each* of the markup language *section properties of the application document*, wherein *individually* mapping the properties includes:
>
>> setting an option element in the mini-document markup language element, wherein the option element includes at least one member of a group comprising: a header value and a footer value,
>>
>> setting a type attribute in the mini-document markup language element, wherein the type attribute includes a value that indicates an occurrence pattern of the body of the mini-document within the application document,
>>
>> setting page size properties of the application document in the section properties of the application document, wherein the page size properties includes a size value of the page, and
>>
>> setting a margin properties of the application document in the section properties of the application document, wherein the margin properties include a top margin value, a bottom margin value, a left margin value, a right margin value and a position value of the location of the mini-document within the section of the application document;
>
> storing *each of the individually mapped* properties of the mini-document in the markup language document;
>
> *validating the markup language document in accordance with a native schema of the word processing application* having definitions for the mini-document, wherein the definitions for the mini-document include a definition for headers, a definition for footers, a definition for a context free chunk, a definition for a paragraph element, a definition for a table element and a definition for a mini-document type; and

parsing and rendering the markup language document _on an application other than the word processing application, wherein the application other than the word processing application does not have access to the native schema of the word processing application having the definitions of the mini-document,_

wherein the _individually mapped_ option element in the section properties _causes_ the rendering of at least one member of a group comprising, a header according to the header value for the section, and a footer according to the footer value for the section,

wherein the _individually mapped_ type attribute in the section properties _causes_ the body portion of the mini-document to be rendered in accordance with the occurrence pattern of the section, wherein the value is at least one member of a group comprising: an odd page value for the section and an even page value for the section,

wherein the _individually mapped_ page size properties for the section _causes_ the page to be rendered according to the size value of the page of the section, and

wherein the _individually mapped_ margin properties for the section _causes_ the rendering of a top margin according to the top margin value, a bottom margin according to the bottom margin value, a left margin according to the left margin value, a right margin according to a right margin value and a mini-document position according to the position value of the location of the mini-document within the section.

The specification of the current application sets forth the advantages of the above combination of features. The Summary of the application states that the application relates to representing header and footer information in ML so that applications running in environments where the header and footer generation information has not been installed are able to still render the header and footer structures. (Specification at page 2, lines 18-23). The specification continues by stating that the ML document may be read by applications that do not share the same schema that created the document. (Specification at page 5, lines 23-25). The specification further continues by stating that the header and footer structures may be parsed by applications that understand the markup other than the application that generated the ML file. (Specification at page 6, lines 21-22). With regard to the properties of the mini-document, the

specification states that the mini-document information used within a document may include different headers and footers, including those that are not natively supported by later applications parsing the document. (Specification at page 18, lines 8-10). The mapping of the properties as indicated in the claims allows an application other than the word processing application that generated the document to parse and render the markup language document. The parsing and rendering application does not need access to the native schema of the word processing document to parse and render the document. Such a mapping of the properties provides for versatility of documents generated in association with the word processing application. The mapping configuration facilitates the accessibility of the document on a wide range of applications that do not have access to the native schema or the application functionality that generated the footers or headers. This mapping allows for a rich end document for a user receiving the markup language document.

Independent claim 1 has been amended to clarify that the properties of the mini-document are ***individually mapped***. Claim 1 further indicates each of the individually mapped properties of the mini-document are stored in the mark-up language document. Claim 1 further indicates how each of the individually mapped properties, elements and attributes contribute to the rendering of the document. Applicants cannot find any teaching or suggestion in any of the cited references of the individual mapping as indicated. Moreover, independent claim 1 recites: "***parsing and rendering the markup language document on an application other than the word processing application, wherein the application other than the word processing application does not have access to the native schema of the word processing application having the definitions of the mini-document.***" The Office Action agrees that none of the other three references asserted against claim 1 teach or suggest the feature. In the current Office Action, the Jelliffe reference is asserted as teaching this feature. This feature is not taught in Jelliffe. The Office Action cites to page 4 of Jelliffe which states that "[w]hen a DTD is 'very open' or extensible by arbitrary cut-and-paste, only the always-required can be salvaged from the original content model as originally designed.... This is likely to be the case with XML document moved into word processors. We want to be able to retain as much validatability as possible, but simply enough to be implemented and understood, with known behavior." Nowhere in the

Jelliffe document does Jelliffe suggest that the markup language document (as indicated in claim 1) can be parsed and rendered by an application not having access to the native schema of the word processing application that has the mini-document definitions. Jelliffe is teaching that many markup language models contain different operators. Depending on the operators used, the schema for validating the markup language may cause combinatorial explosions. Jelliffe is proposing a weak validation schema to facilitate the validation using various operators. (See Jelliffe at page 1). Jelliffe discloses that this is useful when different methods are being used to generate the markup language document. Jelliffe is silent on the feature of *parsing and rendering the markup language document on an application other than the word processing application, wherein the application other than the word processing application does not have access to the native schema of the word processing application having the definitions of the mini-document*" Accordingly, reconsideration is respectfully requested.

Independent claim 10 includes the following combination of features that is not taught or otherwise suggested by the cited references:

> determining properties relating to a mini-document used within *a word-processing document generated on a word-processing application*, wherein the mini-document includes a body portion having text;

> determining whether the mini-document is at least one member of a group comprising: a header and a footer;

> *individually writing the properties into each of the section properties markup language elements* associated with the word processing document, wherein individually writing the properties includes:

>> writing an option element in the mini-document markup language element, wherein the option element includes at least one member of a group comprising: a header value and a footer value,

>> setting a type attribute, wherein the type attribute includes a value that indicates an occurrence pattern of the body of the mini-document within the application document, wherein upon rendering the markup language document, the type attribute causes the body portion of the mini-document to be repeated in the application in accordance with the occurrence pattern, and

setting a margin properties of the application document in the section properties of the application document, wherein the margin properties include a numerical position value of the location of the mini-document within the section of the word-processing document;

storing each of the individually written properties in the markup language document;

validating the markup language document in accordance with a native schema of the word processing application having definitions for the mini-document; and

parsing and rendering the markup language document *on an application other than the word processing application, wherein the application other than the word processing application does not have access to the native schema of the word processing application having the definitions of the mini-document, wherein the markup language document is rendered according to the properties individually written to the section properties markup language elements*.

Independent claim 10 recites: "*parsing and rendering the markup language document on an application other than the word processing application, wherein the application other than the word processing application does not have access to the native schema of the word processing application having the definitions of the mini-document, wherein the markup language document is rendered according to the properties written to the section properties markup language elements.*" The Office Action agrees that none of the other three references asserted against claim 10 teach or suggest the feature. In the current Office Action, the Jelliffe reference is asserted as teaching this feature. This feature is not taught in Jelliffe. The Office Action cites to page 4 of Jelliffe which states that "[w]hen a DTD is 'very open' or extensible by arbitrary cut-and-paste, only the always-required can be salvaged from the original content model as originally designed.... This is likely to be the case with XML document moved into word processors. We want to be able to retain as much validatability as possible, but simply enough to be implemented and understood, with known behavior." Nowhere in the Jelliffe document does Jelliffe suggest that the markup language document (as indicated in claim 1) can be parsed and rendered by an application not having access to the native schema of the word processing application that has the mini-document definitions. Jelliffe is teaching that many

14

markup language models contain different operators. Depending on the operators used, the schema for validating the markup language may cause combinatorial explosions. Jelliffe is proposing a weak validation schema to facilitate the validation using various operators. (See Jelliffe at page 1). Jelliffe discloses that this is useful when different methods are being used to generate the markup language document. Jelliffe is silent on the feature of *"parsing and rendering the markup language document on an application other than the word processing application, wherein the application other than the word processing application does not have access to the native schema of the word processing application having the definitions of the mini-document, wherein the markup language document is rendered according to the properties written to the section properties markup language elements"* Accordingly, reconsideration is respectfully requested.

Independent claim 18 includes the following combination of features that is not taught or otherwise suggested by the cited references:

> a processor; and
>
> a memory associated with computer-executable instructions configured to:
>
>> determine properties relating to a mini-document *included in at least one section of a word processing application document generated on a word processor*, wherein the mini-document includes a body portion having text;
>>
>> determine whether the mini-document is at least one member of a group comprising: a header and a footer;
>>
>> *individually map the properties into a markup language element that is stored with markup language section properties of the sections of the application document*, wherein individually mapping the properties includes:
>>
>>> setting an option element in the mini-document markup language element, wherein the option element includes at least one member of a group comprising: a header value and a footer value,

15

> setting a type attribute, wherein the type attribute includes a value that indicates an occurrence pattern of the body of the mini-document within the application document,

> setting a margin properties of the application document in the section properties of the application document, wherein the margin properties include a position value of the location of the mini-document within the section of the application document, and

> store each of the individually mapped properties in the markup language section properties of the application document;

> a validation engine configured to validate the markup language document; and

> an application *other than the word processing application, wherein the application other than the word processing application does not have access to a native schema of the word processing application having the definitions of the mini-document, wherein the markup language document is parsed and rendered by the application other than the word processing application according to the properties individually written to the section properties markup language elements.*

Independent claim 18 recites: "*an application other than the word processing application, wherein the application other than the word processing application does not have access to a native schema of the word processing application having the definitions of the mini-document, wherein the markup language document is parsed and rendered by the application other than the word processing application according to the properties written to the section properties markup language elements.*" The Office Action agrees that none of the other three references asserted against claim 18 teach or suggest the feature. In the current Office Action, the Jelliffe reference is asserted as teaching this feature. This feature is not taught in Jelliffe. The Office Action cites to page 4 of Jelliffe which states that "[w]hen a DTD is 'very open' or extensible by arbitrary cut-and-paste, only the always-required can be salvaged from the original content model as originally designed…. This is likely to be the case with XML document moved into word processors. We want to be able to retain as much validatability as possible, but simply enough to be implemented and understood, with known behavior." Nowhere in the Jelliffe document does Jelliffe suggest that the markup language document (as

indicated in claim 1) can be parsed and rendered by an application not having access to the native schema of the word processing application that has the mini-document definitions. Jelliffe is teaching that many markup language models contain different operators. Depending on the operators used, the schema for validating the markup language may cause combinatorial explosions. Jelliffe is proposing a weak validation schema to facilitate the validation using various operators. (See Jelliffe at page 1). Jelliffe discloses that this is useful when different methods are being used to generate the markup language document. Jelliffe is silent on the feature of *"an application other than the word processing application, wherein the application other than the word processing application does not have access to a native schema of the word processing application having the definitions of the mini-document, wherein the markup language document is parsed and rendered by the application other than the word processing application according to the properties written to the section properties markup language elements"* Accordingly, reconsideration is respectfully requested.
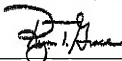
With regard to the dependent claims, the dependent claims include features that are not taught or otherwise suggested by the cited references. Furthermore, those claims ultimately depend from the independent claims above. As such, they should be found allowable for at least those same reasons.

IV.   Request for Reconsideration

In view of the foregoing amendments and remarks, all pending claims are believed to be allowable and the application is in condition for allowance. Therefore, a Notice of Allowance is respectfully requested. Should the Examiner have any further issues regarding this application, the Examiner is requested to contact the undersigned attorney for the applicants at the telephone number provided below.

Respectfully submitted,

MERCHANT & GOULD P.C.

RYAN T. GRACE

Registration No. 52,956
Direct Dial: 402.344.3000

MERCHANT & GOULD P.C.
P. O. Box 2903
Minneapolis, Minnesota 55402-0903
612.332.5300

27488
PATENT TRADEMARK OFFICE